CHOUETTE

Maintenance, accompagnement et recette de logiciels pour les échanges de données multimodales

Manuel utilisateur de l'application CHOUETTE en ligne de commandes

CHOUETTE V2.3

Janvier 2013

Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques

Informations sur ce document :

<table>
<tr><td colspan="4">Organisme commanditaire : AFIMB</td></tr>
<tr><td colspan="4">Titre : Manuel utilisateur de l'application CHOUETTE en mode lignes de commande</td></tr>
<tr><td colspan="4">Sous-titre :</td></tr>
<tr><td colspan="2">Organismes auteurs
CITYWAY
MOBIGIS
CETE Med.

Maitre d'ouvrage
DGITM/AFIMB</td><td>Rédacteurs
Marc FLORISSON
Zakaria BOUZIANE
Michel ETIENNE
Luc DONNET
Patrick Hérard
Cyril BLOUDEAU
Olivier LEMETTAIS
Hugo CHAUVIN
Jérémie GASTON-RAOUL</td><td>Participants
Patrick GENDRE
Marc FLORISSON
Zakaria BOUZIANE
Michel ETIENNE
Luc DONNET
Yannick DENIS
Frédéric SCHETTINI
Laurent DEZOU
Jean-Dominique LECA</td></tr>
<tr><td colspan="4">Résumé :

CHOUETTE est un logiciel libre développé à l'initiative du ministère français chargé des transports (et du développement durable), dans le but de faciliter l'échange de données d'offre (théorique) de transport collectif (TC), en s'appuyant pour cela sur la norme NFP 99506, dite Neptune, qui spécifie un profil d'échange XML.

Les utilisateurs visés sont les collectivités locales Autorités Organisatrices de Transport (AOT), les exploitants des réseaux TC ,et leurs prestataires (bureaux d'étude ou société de services). D'autres utilisateurs potentiels sont néanmoins identifiés : services de l'état, éditeurs de logiciels, opérateurs de services d'information, chercheurs...
Le présent document décrit l'utilisation de l'application CHOUETTE en mode commande (sans navigateur web).</td></tr>
<tr><td colspan="2">Mots clés :
profil d'échange Neptune, information multimodale, application Chouette, manuel utilisateur, IHM</td><td colspan="2">Diffusion :

publique (licence Creative Commons CC-by-nd)</td></tr>
<tr><td>Nombre de pages :
14 pages</td><td>Date
Janvier 2013</td><td>Confidentialité :
Non</td><td>Bibliographie :
Oui</td></tr>
</table>

Historique des versions / révisions :

Historique des versions / révisions			
Version/ révision	Date d'application	Description des changements	Auteur
V2.0	21/01/13	Nouveau document issu à partir du manuel Chouette 1.7	Patrick GENDRE
V2.3	23/04/14	Changement du titre du doc de V2.0 vers V2.3	Pia MURGAT

Sommaire

1.Chouette – Mode commande....4

1.1 Présentation....4

1.2 Installation....4

1.3 Modes d'activation....4

1.4 Options générales....4

1.5 Commandes disponibles....5

1.5.1Commandes d'échanges de données....5

1.5.2Commandes de manipulation des objets....6

1.5.3Commande d'accès à la base....7

1.5.4Autres commandes....7

1.6 Exemples....8

1.7 Import et export en ligne de commande....11

1.7.1.1Import Neptune (Ligne)....11

1.7.1.2Import CSV (Ligne)....11

1.7.1.3Import GTFS (Ligne)....11

1.7.1.4Export NEPTUNE (Ligne)....12

1.7.1.5Export CSV (Ligne)....12

1.7.1.6Export GTFS (Ligne)....12

1.7.1.7Export Géoportail (Réseau)....12

1. Chouette – Mode commande

1.1 Présentation

Afin de réaliser des actions automatiques ou d'agir interactivement sur les données Neptune sans passer par l'application WEB de Chouette, un mode commande est disponible sous la forme d'une application JAVA lancer depuis une console Windows ou un terminal Linux.

L'application s'exécute en local sur la machine sur laquelle est est installée, elle n'est pas accessible à distance et dans le cas d'utilisation des commandes agissant sur la base de données, la machine doit pouvoir se connecter à celle-ci.

1.2 Installation

L'archive java chouette-command.zip est disponible en téléchargement sur le site internet www.chouette.mobi dans le menu CHOUETTE / Version précompilée : http://www.-chouette.mobi/spip.php?rubrique87 .

Merci d'avance de nous laisser vos coordonnées via le site chouette.mobi pour que nous ayons connaissance de nos utilisateurs !

Il est également possible de générer ce fichier zip à partir du code avec Maven : pour cela, reportez-vous au manuel d'installation.

1.3 Modes d'activation

Chouette-command dispose de 2 scripts de lancement :

- chouette.bat : pour Windows
- chouette.sh : pour Linux

Chouette-command peut exécuter un enchainement de commandes élémentaires sous 3 modes :

- commandes passées par argument
  - `chouette[.sh] [options générales] -c commande1 [arguments] -c commande2 [arguments] ...`
- commandes fournies dans un fichier script
  - `chouette[.sh] [options générales] -f fichier_script`
- commandes saisies interactivement par l'opérateur
  - `chouette[.sh] [options générales] -i`

En dehors du mode interactif; le processus d'exécution des commandes est interrompu dès qu'une de celles-ci retourne une erreur.

Note : sous Windows, il convient de passer la console sur la fonte de caractères LUCIDA Console et une fois activée, de lancer la commande '`chcp 1252`' pour visualiser les codes UTF-8

1.4 Options générales

Les options générales de chouette-command sont :

- -noDao : n'active pas les modules d'accès à la DAO

  cette option *mise impérativement en premier* permet d'alléger le lancement de l'application dans le cas d'un usage ne nécessitant pas la base de donnée, comme par

exemple des conversions de format ou une validation de fichiers Neptune. Il n'est pas nécessaire de disposer non plus de la base de données dans ce cas.

Attention : dans le cas où cette option n'est pas utilisée, l'application Chouette va se connecter à la base de donnée ; il est impératif que le schéma ait été créé par l'application WEB, dans le cas contraire, celui-ci ne pourra pas être exploité par celle-ci.

- -**v**erbose : affiche des informations lors de l'exécution des commandes comme par exemple le numéro d'ordre d'exécution de la commande, le temps d'exécution, les paramètres, ...
- -**h**elp [command] : affiche l'aide générale des commandes disponibles ou le détail d'une commande spécifiée
- -**o**bject [neptuneObjectName] : précise sur quel type d'objet les commandes suivantes vont devoir s'exécuter.

  Cette option est aussi disponible après chaque commande pour changer de type d'objet.
- -**i**nteractive : active le mode interactif, les commandes seront saisie sur la console ; le "prompt" précisera à chaque fois quel est le type d'objet actif.
- -**f**ile [fileName] : active le mode script, les commandes sont lues dans le fichier script

  Le fichier script doit contenir une commande par ligne; des lignes blanches ou commençant par le caractère # (dièse) sont ignorées et la commande quit permet d'interrompre le script avant la fin du fichier (toutefois, les lignes au delà de cette commande doivent être syntaxiquement correctes).
- -**c**ommand [command] [command arguments] : exécute la commande fournie avec ses arguments.

  Plusieurs blocs d'arguments commençant par -**c**ommand permettent d'enchainer les commandes sans écrire un script ni passer en mode interactif.

1.5 Commandes disponibles

Rappel : l'option -object [neptuneObjectName] est disponible sur chaque commande pour changer le type de donnée Neptune si nécessaire; le changement de type efface de la mémoire les objets précédemment chargés.

1.5.1 Commandes d'échanges de données

- getImportFormats : fournit la liste des modules d'import de données disponibles pour le type d'objet actif,

  cette méthode interroge les modules d'import pour retourner aussi les paramètres spécifiques attendus.
- import : importe les données à un format connu

  options :

  -format [formatName] : précise le format (nom fourni par getImportFormats)

  -optionN [valeurOptionN] : renseigne la valeur d'un des paramètres de l'import

  optionN représente le nom du paramètre tel qu'il est retourné par getImportFormats

  valeurOptionN représente la valeur du paramètre (optionnel pour un booléen où la présente de l'option revient à une valeur "true"

en retour de cette commande, le modèle importé est en mémoire du programme en attente d'une nouvelle commande (sauvegarde dans la base, validation, export à un autre format, ...)

- getExportFormat : fournit la liste des modules d'export de données disponibles pour le type d'objet actif (comportement identique à getImportFormats)
- export : exporte les données disponibles en mémoire (options identiques à import)
- getDeletionExportFormats : fournit la liste des modules d'export de de suppression données disponibles pour le type d'objet actif (comportement identique à getImport-Formats)
- exportForDeletion : exporte la demande de suppression des données disponibles en mémoire (options identiques à import)

  Note : l'export de suppression doit être effectué avant la suppression effective des données de la base.
- validate : valide les objets en mémoire

  options :

  -file [filename] : fichier d'enregistrement du rapport de validation (par défaut le rapport est imprimé à la console)

  -append : (optionnel) ajoute le rapport au fichier (en l'absence de cette option, le précédent contenu du fichier est effacé)
- complete : enrichit les attributs des objets en mémoire dans le but de les exporter.

  Note : les objets lus dans la base de données ne contiennent par défaut que les informations utiles à leur mise à jour; les modules d'export peuvent avoir besoin de données complémentaires comme les identifiants métier des objets en relation directe ou indirecte avec l'un d'entre eux , cette commande résout l'ensemble de ces données.

  Exemple : LineIdShortcut dans l'objet VehicleJourney

1.5.2 Commandes de manipulation des objets

Ces commandes (à l'exception de new et print) ne sont disponibles que si un unique objet est en mémoire

- new : crée un nouvel objet en mémoire
- set : fixe la valeur d'un attribut de cardinalité 1 de l'objet en mémoire

  options :

  -attr [attributeName] [value] : ajoute une valeur de type élémentaire

  attributeName = nom de l'attribut

  value = valeur à fixer (entre " " si la valeur contient des espaces); non renseignée pour supprimer la valeur existante.

  Note : si l'attribut est de type date, heure ou horodate, le format de la valeur doit respecter la syntaxe suivante :

type	format
date	yyyy-MM-dd
heure	HH:mm:ss
horodate	yyyy-MM-dd_HH:mm:ss

  -ref [attributeName] [referenceObjectId] : fixe une référence à un autre objet Neptune (celui-ci doit déjà être présent dans la base)

  attributeName = nom de l'attribut

referenceObjectId = référence Neptune de l'objet (champs ObjectId) ; non renseigné pour supprimer la référence existante.

- add : ajoute une valeur à un attribut de cardinalité multiple (voir set pour la syntaxe, mais les paramètres value ou referenceObjectId sont obligatoires)
- remove : retire une valeur d'un attribut de cardinalité multiple (voir set pour la syntaxe, mais les paramètres value ou referenceObjectId sont obligatoires)
- print : imprime les attributs et les dépendances d'un objet

  options :

  -**l**evel [deepness] : profondeur d'affichage des objets en dépendance (par défaut 0 = uniquement les attributs directs)

1.5.3 Commande d'accès à la base

- get : charge des objets depuis la base

  options :

  -id [value]+ : filtre sur une liste d'identifiants techniques (clé primaire de la base)

  -objectId [value]+ : filtre sur une liste d'identifiants métier (ObjectId)

  -filter [filterargs]+ : filtre sur une requête sur attribut

  filterargs = [attribute] [operator] [value]

  attribute = nom de l'attribut (peut être l'attribut d'un sous objet ou d'une sous structure , comme par exemple ptnetwork.objectId sur un objet de type Line)

  operator = [null | eq | = | like]

  value (si nécessaire) = valeur de comparaison
- save : sauve les objets en mémoire dans la base

  options :

  -propagate : propage la sauvegarde aux objets liés (issus d'un import par exemple)
- delete : supprime les objets en mémoire de la base ainsi que les objets en dépendance structurelles. (la liste des objets supprimés en cascade est définie dans le document d'architecture)

  options :

  -propagate : supprime aussi les objets liés mais pouvant avoir une existence indépendante

  exemple : lors d'une suppression de ligne , suppression des arrêts du référentiel qui ne sont pas partagés avec une autre ligne. (la liste des objets supprimés par propagation est définie dans le document d'architecture)

1.5.4 Autres commandes

- **q**uit ou exit : arrête le script ou le mode interactif
- verbose : active ou désactive le mode verbeux

  options :

  -on (par défaut) : active le mode verbeux

  -off : désactive le mode verbeux
- lang : change la langue des messages

  options :

  -fr : passage en français

-en : passage en anglais

- info : fournit la liste des attributs d'un objet
- help : fourni l'aide sur la liste des commandes disponibles ou le détail sur une commande précisée

  options :

  -cmd [commandName] : aide détaillée sur la commande
- include : insère le contenu d'un autre fichier script dans le fichier en cours de chargement

  options :

  -**f**ile [filename] : chemin du fichier script à inclure (chemin absolu ou relatif à l'emplacement du fichier contenant l'instruction include)
- exec : exécute les commandes d'un fichier depuis le mode interactif

  options :

  -**f**ile [filename] : chemin du fichier script à exécuter

1.6 Exemples

1. commandes passées par argument

import Neptune + sauvegarde en base

```
chouette[.sh] -c import -o line -format NEPTUNE -inputFile reseau.zip -validate -c save -propagate
```

(.sh est nécessaire sous linux)

import Neptune + validation :

```
chouette[.sh] -noDao -c import -o line -format NEPTUNE -inputFile reseau.zip -validate -c validate -f report.txt
```

export Neptune :

```
chouette[.sh] -c get -o line -filter ptnetwork.objectId eq 'NINOXE:GroupOfLine:tatrobus' -c export -format NEPTUNE -outputFile tatrobus.zip
```

2. commandes écrites dans un script

script de création d'objet :

```
# création d'une nouvelle instance de réseau
new -o ptnetwork
# valorisation des attributs
set -attr objectId NINOXE:GroupOfLine:Test
set -attr objectVersion 1
set -attr versionDate 2011-08-30
set -attr name "Réseau de démonstration"
# sauvegarde en base
save
#  création d'une nouvelle instance de transporteur
new -o company
```


Certu - 2013

```
set -attr objectId NINOXE:Company:TransDemo
set -attr objectVersion 1
set -attr name "Transporteur de démonstration"
save
# création d'un arrêt commercial
new -o StopArea
set -attr objectId NINOXE:StopArea:ARR1
set -attr objectVersion 1
set -attr name "Arrêt numéro 1"
set -attr areaType CommercialStopPoint
# géolocalisation
set -attr areacentroid.objectId NINOXE:AreaCentroid:ARR1
set -attr areacentroid.objectVersion 1
set -attr areacentroid.longitude 4.040980161539514
set -attr areacentroid.latitude 49.21905234063518
set -attr areacentroid.longLatType WGS84
set -attr areacentroid.projectedPoint.x 724260
set -attr areacentroid.projectedPoint.y 2470377
set      -attr      areacentroid.projectedPoint.projectionType
epgs:27572
set -attr areacentroid.name "Arrêt numéro 1"
save
# création d'une ligne
new -o line
set -attr objectId NINOXE:Line:1
set -attr objectVersion 1
set -attr name "LIGNE 1"
set -attr number 1
set -attr publishedName "LIGNE 1"
set -attr transportModeName Bus
# association au réseau
set -ref ptNetwork NINOXE:GroupOfLine:Test
# association au transporteur
set -ref company NINOXE:Company:TransDemo
set -attr mobilityRestrictedSuitable false
save
```

utilisation du script :

```
chouette[.sh] -f script.txt
```

3. mode interactif :

lancement :

```
chouette[.sh] -f script.txt
```

déroulement :

```
unknown object (0) >info -o line
Command 1 : info
    parameters object : [line]
- objectId : type String
- objectVersion : type int
- creationTime : type Date
- creatorId : type String
- name : type String
- number : type String
- publishedName : type String
- transportModeName : type TransportModeNameEnum
       Air Train LongDistanceTrain LongDistanceTrain_2 Local-
Train RapidTransit Metro
       Tramway Coach Bus Ferry Waterborne PrivateVehicle Walk
Trolleybus Bicycle
     Shuttle Taxi VAL Other
- registrationNumber : type String
- comment : type String
- ptNetworkIdShortcut : type String
- lineEnd : collection of type String (add allowed)
- ptNetwork : type PTNetwork
- company : type Company
- companie : collection of type Company
- route : collection of type Route (add allowed) (remove allo-
wed)
- mobilityRestrictedSuitable : type Boolean
- userNeed : collection of type UserNeedEnum (add allowed)
     allergic heartCondition otherMedicalNeed visualImpairment
auditoryImpairment
     cognitiveInputImpairment averseToLifts averseToEscalators
     averseToConfinedSpaces averseToCrowds otherPsychosensory-
Need luggageEncumbered
     pushchair baggageTrolley oversizeBaggage guideDog otherA-
nimal otherEncumbrance
       wheelchair assistedWheelchair motorizedWheelchair wal-
kingFrame
     restrictedMobility otherMobilityNeed
- groupOfLine : type GroupOfLine
- groupOfLine : collection of type GroupOfLine
- restrictionConstraint : collection of type RestrictionCons-
traint (add allowed) (remove allowed)
- facilitie : collection of type Facility
command info executed in 94 ms
line (0) > new
Command 3 : new
```


CC BY ND

```
line (1) > q
```

fin d'exécution

1.7 Import et export en ligne de commande

Chaque module d'import dispose de ses propres arguments; dans ce paragraphe, ne sont présentés que ceux disponibles en standard

Les paramètres des modules d'import/export sont toujours disponibles en utilisant les commandes getImportFormats et getExportFormats

1.7.1.1 Import Neptune (Ligne)

Paramètre	type	obligatoire	valeur par défaut	extensions (fichier)	description
inputFile	chemin de fichier	oui	s.o.	xml zip	chemin du fichier à importer
fileFormat	texte	non *			type de fichier si l'extension n'est pas conforme au format
validate	booléen	non	false		active la validation des données selon la XSD Neptune
optimizeMemory	booléen	non	false		Optimise la gestion mémoire pour les imports volumineux (ralentit le traitement)

1.7.1.2 Import CSV (Ligne)

Paramètre	type	obligatoire	valeur par défaut	extensions (fichier)	description
inputFile	chemin de fichier	oui	s.o.	csv	chemin du fichier à importer
fileFormat	texte	non *			type de fichier si l'extension n'est pas conforme au format
objectIdPrefix	texte	non	paramètre system.-code de la configuration		fournit le préfixe des identifiants Neptune des objets générés

1.7.1.3 Import GTFS (Ligne)

Paramètre	type	obligatoire	valeur par défaut	extensions (fichier)	description
inputFile	chemin de fichier	oui	s.o.	zip	chemin du fichier à importer
fileFormat	texte	non *			type de fichier si l'extension n'est pas conforme au format
objectIdPrefix	texte	non	paramètre system.-code de la		fournit le préfixe des identifiants Neptune des objets générés

			configuration		
maxDistanceForCommercial	numérique	non	10		Distance maximale en mètre entre 2 arrêts de même nom pour les regrouper au sein d'un arrêt commercial
ignoreLastWord	booléen	non	false		Ignorer le dernier mot du nom de l'arrêt lors des regroupements
ignoreEndChars	numérique	non	0		Nombre de caractères à ignorer en fin du nom de l'arrêt lors des regroupements
maxDistanceForConnectionLink	numérique	non	50		Distance maximale pour créer une correspondance entre 2 arrêts voisins et non regroupés dans un arrêt commercial
optimizeMemory	booléen	non	false		Optimise la gestion mémoire pour les imports volumineux (ralentit le traitement)

1.7.1.4 Export NEPTUNE (Ligne)

Paramètre	type	obligatoire	valeur par défaut	extensions (fichier)	description
outputFile	chemin de fichier	oui	s.o.	xml,zip	chemin du fichier à produire

1.7.1.5 Export CSV (Ligne)

n'accepte qu'une ligne ne possédant qu'au maximum une séquence d'arrêt aller et une retour

Paramètre	type	obligatoire	valeur par défaut	extensions (fichier)	description
outputFile	chemin de fichier	oui	s.o.	csv	chemin du fichier à produire

1.7.1.6 Export GTFS (Ligne)

Paramètre	type	obligatoire	valeur par défaut	extensions (fichier)	description
outputFile	chemin de fichier	oui	s.o.	xml,zip	chemin du fichier à produire
timeZone	texte	oui	s.o.		fuseau horaire selon le codage TZ (voir http://fr.wikipedia.org/wiki/Tz_database)

1.7.1.7 Export Géoportail (Réseau)

Paramètre	type	obligatoire	valeur par	extensions	description

Certu - 2013

			défaut	(fichier)	
outputFile	chemin de fichier	oui	s.o.	xml,zip	chemin du fichier à produire
logoFile	chemin de fichier	oui	s.o.	gif, png, jpg, jpeg, tmp	fichier image du logo de l'AOT
logoFileName	texte	non	nom du fichier logoFIle		nom du fichier s'il arrive par upload
aotURL	texte	oui	s.o.		URL de l'AOT
legalInforma-tion	texte	non	vide		Informations légales
legalInforma-tionURL	texte	non	vide		URL des informations légales
aotAddress	texte	oui	s.o.		Adresse de l'AOT
aotEmail	texte	oui	s.o.		Email de l'AOT
aotPhone	texte	oui	s.o.		Numéro de téléphone de l'AOT
readMe	texte	oui	s.o.		Contenu du fichier ReadMe.txt
stopNote	texte	oui	s.o.		Notes sur les arrêts
accessNote	texte	oui	s.o.		Note sur les accès
quayPicto	chemin de fichier	non		png	Icône de représentation d'un quai (50x50)
quayPicto-MinScale	numé-rique	non	2000		Échelle d'affichage minimale de l'icône
quayPicto-MaxScale	numé-rique	non	4000		Échelle d'affichage maximale de l'icône
boardingPosi-tionPicto	chemin de fichier	non		png	Icône de représentation d'un quai (50x50)
boardingPosi-tionPicto-MinScale	numé-rique	non	2000		Échelle d'affichage minimale de l'icône
boardingPosi-tionPicto-MaxScale	numé-rique	non	4000		Échelle d'affichage maximale de l'icône
commercial-StopPointPicto	chemin de fichier	non		png	Icône de représentation d'un quai (50x50)
commercial-StopPointPic-toMinScale	numé-rique	non	1600		Échelle d'affichage minimale de l'icône
commercial-StopPointPic-toMaxScale	numé-rique	non	4000		Échelle d'affichage maximale de l'icône

stopPlacePicto	chemin de fichier	non		png	Icône de représentation d'un quai (50x50)
stopPlacePic-toMinScale	numé-rique	non	1600		Échelle d'affichage minimale de l'icône
stopPlacePic-toMaxScale	numé-rique	non	4000		Échelle d'affichage maximale de l'icône
accessPoint-Picto	chemin de fichier	non		png	Icône de représentation d'un quai (50x50)
accessPoint-PictoMinScale	numé-rique	non	1000		Échelle d'affichage minimale de l'icône
accessPoint-PictoMaxS-cale	numé-rique	non	2000		Échelle d'affichage maximale de l'icône

CC BY ND